

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>1.Abend</p> <ul style="list-style-type: none"> Sie können grundlegende UNIX-Befehle erklären und deren Bedeutung für die Python-Entwicklung beschreiben (K2). Sie können den Aufbau einer Entwicklungsumgebung für Python auf einem BYOD-Gerät erklären (K2). 			
<p>Vorstellung (Wer bin ich? Problem-Based Learning)</p> <p>Installieren der Entwicklungsumgebung</p> <ul style="list-style-type: none"> VNC / Notepad++ / PuTTY / FTP Client Nano auf RPi, die wichtigsten Befehle 1.Programm <ul style="list-style-type: none"> File erstellen und editieren (Console, Nano) print(), #! /usr/bin/pyhton3 LINUX Befehle: <ul style="list-style-type: none"> cd, ls -al, pwd, rm, mv, cp, mkdir, LINUX file-system (chmod, filepath) Execution Fehlermeldungen interpretieren können und Lösungen implementieren Notepad++, Putty Programm erweitern (Print(), String-Operationen, Input()) <p>Aufgabe 1a (Umrechner.py)</p> <ul style="list-style-type: none"> Menu (f-Strings, Multiline-Strings) User-Input (Wähle:) If-then-elif-else Struktur Loop mit 0 beenden Behandlung von falsch Eingaben Formeln implementieren (Variablen, Float-Input, Math-Operationen) 	<p>Nach Anleitung installieren und konfigurieren</p> <p>Test-Driven Approach mit Reviews und Refactoring</p> <p>Theoretischen Einschüben</p>	<p>10'</p> <p>60'</p> <p>130'</p>	<p>Umrechner.py Menu implementieren</p>

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>2.Abend</p> <ul style="list-style-type: none"> Sie können einfache Python-Skripte schreiben, um Daten zu verarbeiten und Funktionen effizient einzusetzen (K3). Sie können API-Dokumentationen analysieren und daraus ableiten, wie externe Pakete in eigene Anwendungen integriert werden können (K4). 			
<p>Aufgabe 1b (Umrechner.py)</p> <ul style="list-style-type: none"> halt() implementieren import math Formeln in Funktionen implementieren Funktionen abwärtskompatible erweitern Exception Handling mit Pre-Checks und try-catch 	<p>Test-Driven Approach mit Reviews und Refactoring</p> <p>Theoretischen Einschüben</p>	100'	
<p>Aufgabe 1c (Umrechner.py, xxLibrary.py)</p> <ul style="list-style-type: none"> Funktion in eigene Library auslagern Refactoring Umrechner.py verwendet eigene Library Weitere Funktionen readInt(), readFloat() implementieren, testen, anwenden und in eigene Library übernehmen. Neuer Menu-Punkt: Quadratische Gleichung implementieren (Return container) 	<p>Test-Driven Approach mit Reviews und Refactoring</p> <p>Theoretischen Einschüben</p>	100'	
			Umrechner.py and My_XXX_Library.py fertigstellen

Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
<p>3.Aband</p> <ul style="list-style-type: none"> Sie können verschiedenen Containers für Python kritisch vergleichen und Empfehlungen aussprechen (K5). Sie können JSON- und XML-Daten verarbeiten (K3). 			
<p>Leistungskontrolle 1 (Linux Commands)</p> <p>Elemete in den verschiedenen Containers zugreifen (lesen), hinzufügen/ändern und löschen. Listen[], Tupels(), Dictionaries{}</p> <p>Sub-Listen mit [1:-1] Ranges lesen resp verarbeiten/ändern.</p> <p>for – Loops</p> <ul style="list-style-type: none"> Listen und Tuples Dictionaries (keys()) <p>Comprehensions mit Filter und ZIP für eigene Anwendungen einsetzen können.</p> <p>Umrechner Erweiterungen:</p> <ul style="list-style-type: none"> Quadratische Gleichung Fakultät Primzahlen Rechner Primzahlen und Teiler Listen Filter-Berechnungen 	<p>Formativer Test</p> <p>Test-Driven Approach mit Reviews und Refactoring</p> <p>Theoretischen Einschüben</p>	<p>15'</p> <p>60'</p> <p>140'</p>	<p>Weitere Funktionen für umrechner.py implementieren</p>

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>4.Abend</p> <ul style="list-style-type: none"> Sie können die Funktionen des Moduls Sense_Hat anhand der API Dokumentation richtig anwenden (K4). 			
<p>Linien Aufgaben lösen</p> <p>Aufgabe 2a (LED_Matrix.py)</p> <ol style="list-style-type: none"> setPixel(), setPixels(), clear(), sleep(), showMessage() Eventhandling (Joystick) IMU- und Meteo-Sensoren <p>Aufgabe 2b (xx_SenseHat_Librarie.py)</p> <ol style="list-style-type: none"> setPixel() mit clipping drawLine(), drawRecantgle(), drawCircle() Functions erweitern mit fillColor und borderColor drawCompassNeedle(azimutInGrad) 	<p>SOL</p> <p>Test-Driven Approach mit Reviews und Refactoring</p> <p>Theoretischen Einschüben</p>	<p>90'</p> <p>110'</p>	<p>Sense_Hat Library mit setPixel(), drawLine(), drawCircle(), draw-Rectangle()</p>

SC\Schulungsunterlagen\HBU\03_PYT1\Unterrichtsplan_HFE_PYT1.docx

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>6.Abend</p> <ul style="list-style-type: none"> Sie können REST-Services in Python aufrufen und deren Respons-Daten (JSON, XML,...) in einer Anwendung verarbeiten (K3). Sie können Filehandling-Mechanismen anwenden und bewerten (K5). 			
Open-Weather REST Service mit eigenem Token (AppID) aus Python aufrufen (requesten) und response als JSON Struktur verarbeiten.	Test-Driven Approach mit Reviews und Refactoring	40'	Design und Implementation einer Wetter-Logger (Wetterstation), welche Meteo-Daten von einem Ort / Lokation in ein File schreibt
Filehandling open() for read, write and append (inkl UTF and ASCII)	Theoretischen Einschüben	40'	
		120'	

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>7.Aband</p> <ul style="list-style-type: none"> Sie können die Vor- und Nachteile objektorientierter Programmierung in Python bewerten (K5). Sie können eine eigene Klasse designen und diese modular in einer Anwendung nutzen (K3). 			
<p>Eine eigene, einfache Logger-Klasse mit log_msg() ohne Strategien entwickeln und testen.</p> <p>Anschliessend eigene Logger-Klasse in Meteo-App einsetzen.</p>	<p>Test-Driven Approach mit Reviews und Refactoring</p> <p>Theoretischen Einschüben</p>	200'	<p>Eigene einfache Loggerklasse in Wetter_Logger verwenden</p>

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>8.Aband</p> <ul style="list-style-type: none"> Sie können ein Konzept für die Arbeitsteilung in einem Python-Projekt entwickeln und umsetzen (K4). Sie können die Effizienz und Sicherheit verschiedener Methoden für den Zugriff auf Raspberry Pi-Systeme beurteilen (K5). 			
<p><u>Leistungsnachweis (Modullernzielkontrolle MILZ):</u></p> <p>Eine Aufgabe unter Zeitdruck gemäss Spezifikationen nach dem Test-Driven Approach implementieren.</p>	<p>Selbstständiges programmieren und individuellen Review durch Dozenten.</p>	<p>190'</p>	

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
9.Aband <ul style="list-style-type: none">Sie können Bilder und PDFs entwickeln (K4).Sie können unterschiedliche Automatisierungsansätze bewerten und für spezifische Szenarien adaptieren (K5).			
Fachgespräche über MLZ Bildbearbeitung mit Python	Einzelgespräche gemäss Zeitplan Selbstorganisiertes Lernen (SOL)	200‘	

- Bemerkungen:**
- Jeder Abend dauert 4 Lektionen.
 - Der Unterrichtsplan kann bei Bedarf dem vorhandenen Wissen der Klasse angepasst werden.
 - Die Studierenden lösen die Übungen auf ihren privaten Notebooks.
 - Der Leistungsnachweis (**MLZ**) am 8.Aband ist in Einzelarbeit in der vorgegebenen Zeit zu erstellen